

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Yukio OGUMA

Application No.:

Group Art Unit:

Filed: February 26, 2004

Examiner:

For: APPARATUS, METHOD AND RECORDING MEDIUM FOR STARTING UP DATA
PROCESSING SYSTEM

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)
herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-119808

Filed: April 24, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: February 26, 2004

By: 

Randall Beckers
Registration No. 30,358

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 4 月 2 4 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 1 1 9 8 0 8
Application Number:

[ST. 10/C]: [J P 2 0 0 3 - 1 1 9 8 0 8]

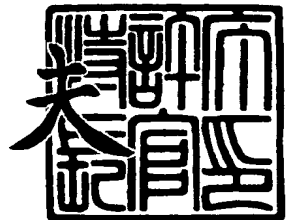
出 願 人 富 士 通 株 式 有 限 公 司
Applicant(s):



2 0 0 3 年 1 2 月 1 6 日

特 許 庁 長 官
Commissioner,
Japan Patent Office

今 井 康



出 証 番 号 出 証 特 2 0 0 3 - 3 1 0 4 4 0 5



【書類名】 特許願

【整理番号】 0350902

【提出日】 平成15年 4月24日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06F 9/445
G06F 1/00

【発明の名称】 情報処理システム

【請求項の数】 5

【発明者】
【住所又は居所】 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通
株式会社内

【氏名】 小熊 幸雄

【特許出願人】
【識別番号】 000005223
【氏名又は名称】 富士通株式会社

【代理人】
【識別番号】 100077517
【弁理士】
【氏名又は名称】 石田 敬
【電話番号】 03-5470-1900

【選任した代理人】
【識別番号】 100092624
【弁理士】
【氏名又は名称】 鶴田 準一

【選任した代理人】
【識別番号】 100100871
【弁理士】
【氏名又は名称】 土屋 繁

**【選任した代理人】****【識別番号】** 100082898**【弁理士】****【氏名又は名称】** 西山 雅也**【選任した代理人】****【識別番号】** 100081330**【弁理士】****【氏名又は名称】** 樋口 外治**【手数料の表示】****【予納台帳番号】** 036135**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【包括委任状番号】** 9905449**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 情報処理システム

【特許請求の範囲】

【請求項 1】 多重化された複数の装置から選択して読み出されたオペレーションシステムが起動されてシステムが立ち上げられる情報処理システムにおいて、

前記複数の装置からブート対象を設定する環境情報を格納する格納手段と、

前記環境情報の設定に基づいてブート対象装置を決定し、該ブート対象装置に格納された前記オペレーションシステムを起動させるブート制御手段と、

前記複数の装置の多重化制御を実行する制御手段と、を有し、

前記制御手段は、前記ブート対象装置に異常が検出されたとき、前記環境情報の設定を変更して、他の前記装置に切り換え制御することを特徴とする情報処理システム。

【請求項 2】 前記環境情報は、

前記複数の装置についてブート対象候補とする装置設定情報を含む第 1 変数情報と、

前記装置設定情報に基づいてブート対象装置を指定するインデックス情報を含む第 2 変数情報と、

前記多重化が有効か否かの 2 値が設定される第 3 変数情報と、を含むことを特徴とする請求項 1 に記載の情報処理システム。

【請求項 3】 前記ブート制御手段は、システムの電源投入時において、前記インデックス情報を初期値にクリアし、前記装置設定情報に最初に設定された前記装置をブート対象として選択することを特徴とする請求項 2 に記載の情報処理システム。

【請求項 4】 前記ブート制御手段は、第 3 変数情報に有効が設定されている場合に、前記インデックス情報により指定されたブート対象装置を選択し、否が設定されている場合には、前記装置設定情報において最初に設定された装置をブート対象として選定することを特徴とする請求項 2 に記載の情報処理システム。

【請求項 5】 前記ブート制御手段は、第 3 変数情報に有効が設定されている場合に、前記インデックス情報におけるインデックスを更新することを特徴とする請求項 4 に記載の情報処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、オペレーティングシステムを格納したブート対象装置を多重化して備えた情報処理システムに関し、特に、ブート対象装置の多重化のためのシステムソフトウェアと連携することにより、主系のブート対象装置で起動できなかった場合に、強制的に従系のブート対象装置に切り換えて、次のブート処理で従系のブート対象装置から起動できる情報処理システムに関する。

【0002】

【従来の技術】

一般に、コンピュータなどの情報処理システムにおいては、起動信号の発生に応答してハードディスクなどのブート対象装置からオペレーティングシステムを起動するように構成されている。このような構成の場合、ブート対象デバイスが壊れたときには、オペレーティングシステムを起動できなくなる。特に、このような障害が、コンピュータで発生すると、特に、このコンピュータがネットワーク上で他のコンピュータに接続されているような場合には、他のコンピュータに対する全てのサービスが停止されてしまうという問題が発生する。

【0003】

一方、最近では、情報処理システムの耐障害性を高める一つの技術として、複数のハードディスクを、RAID (Redundant Arrays of Inexpensive Disks) 構成にして冗長化することが行われている。しかし、この RAID 構成は、ミラーリングやストライピングによって、読み出しデータの信頼性を高める技術であり、システム起動に関わる動作の信頼性を高めるものではない。

【0004】

そこで、ブート対象の装置であるディスク装置を多重化、例えば、2 重化した情報処理システムにおいて、いずれかのディスク装置が破壊されたことを検出し

たとき、スイッチ操作によって、システムに備えられた複数のディスク装置の中から、ブート対象ディスク装置を切り換えるようにして、コンピュータのシステム起動が行われる情報処理システムが開発されている。しかし、この情報処理システムでは、その切り換えのためのスイッチ操作は、管理者などによる手動操作で行われるものであり、しかも、システム起動が、正常に行われたか、失敗したかの判断についても人手によって行われていた。

【0005】

このような情報処理システムでは、ディスク装置が破壊された場合に、再度システムを立ち上げるのに、時間を費やし、システム運用に支障をきたすことになるので、2重化されてディスクの一方が破壊されても、他方のディスクに自動的に切り換えることが行われている。そこで、システム起動に関わる動作の信頼性を十分に高めた情報処理システムが提案されている。

【0006】

この情報処理システムにおいては、図6の構成概念図に示されるように、制御装置1に、OSを格納したディスク装置D1及びD2が2重化して備えられ、制御装置1は、CPUを有する制御部2、不揮発性メモリ3、メインメモリ4を含んでいる。さらに、制御部2には、キーボード、モニタ装置などの周辺装置5が接続される。

【0007】

このような情報処理システムにおける2重化されたディスク装置に対する切り換えの仕方の一例として、情報処理システムのOSを備えた制御部が、ディスク装置のいずれかが破壊されたことを検出したとき、システムに備えられた不揮発性メモリに、イニシャル・マイクロプログラム・ロード（IMPL）ディスク情報と、イニシャル・プログラム・ロード（IPL）ディスク装置情報とを自動的に格納するようにしたものがある（例えば、特許文献1を参照）。この例では、不揮発性メモリにIMPLディスク情報とIPLディスク装置情報が自動的に格納されるので、制御部は、次に格納すべきディスク装置情報を知ることができ、自動的に他の正常なディスク装置に切り換えることができる。

【0008】

また、他の例として、個々に OS を格納し 2 重化されたディスク装置において、該ディスク装置に障害が発生したときに、ディスク装置の障害発生から復旧までの時間を短くしようとした情報処理システムがある（例えば、特許文献 2 を参照）。この例による情報処理システムでは、一方のディスク装置が破壊された場合、システムを一旦終了させ、再立ち上げ前に、他方の破壊されていないディスク装置がマスタとなり、破壊された方のディスク装置がスレーブとなるように、切り換え制御がされ、システム登録がなされる。これにより、システムが再立ち上げされたとき、マスタである破壊されていない方のディスク装置の IPL により、OS が起動され、そして、システムが立ち上がるというものである。

【0009】

さらに、別の例として、OS を個々に格納し 2 重化されたディスク装置を有し、OS を起動する手段と、該 OS の起動完了を検出する手段とを有し、該 OS を起動するための起動信号が発生してから経過時間を計時し、該起動信号が発生してから所定の経過時間内に、該 OS の起動完了が検出されたか否かに基づいて、OS の起動対象となるブート対象ディスク装置の切り換えを制御するようにした情報処理システムがある（例えば、特許文献 3 を参照）

この情報処理システムにおいては、OS の起動完了を自動検出でき、起動信号が発生してから所定の経過時間内に OS の起動完了が検出されたか否かに基づいて、OS の起動対象となるブート対象ディスク装置の切り換えが自動的に制御されるため、人手を介在せずとも、OS を正常に起動できたか否かの判断、および別のブートディスク装置からの OS の再起動を全て自動的に行うことが可能となる。

【0010】

一方、これらのような情報処理システムにおける 2 重化されたブート対象のディスク装置の夫々には、ディスク装置を起動するための種々のプログラムが格納されている。個々のディスク装置において、例えば、ディスクの先頭から、ブート・ブロック・プログラム、OS ロード、OS、システムソフトウェアが、格納され、それ以後に、データが格納されている。

【0011】

情報処理システムの立ち上げ時においては、システム内の不揮発性メモリに格納されていたブートファームウェアがメインメモリに展開され、このブートファームウェアの動作により、ブート対象のディスク装置が選択され、当該ディスク装置に格納されているブート・ブロック・プログラムを読み出し、メインメモリに展開する。このブート・ブロック・プログラムが、ブート対象ディスク装置に格納されたOSローダをメインメモリに展開する。次いで、OSローダがOSをメインメモリに読み出すと、OSが、ブート対象ディスク装置に格納されているシステムソフトウェアをメインメモリに展開させる。このシステムソフトウェアが2重化されたディスク装置の主系と従系のどちらで立ち上がっているかをチェックし、2重化の制御を実行する。このシステムソフトウェアの起動により、OSの通常運用時において、2重化されたディスク装置に対する以降の切り換え処理が実施される。

【0012】**【特許文献1】**

特開平3-229331号公報

【特許文献2】

特開2000-81978号公報

【特許文献3】

特開2002-259130号公報

【0013】**【発明が解決しようとする課題】**

従来の情報処理システムにおいては、システム構成装置を2重化することにより、運用時にシステム構成装置にアクセスするときに発生するハードエラーに対して冗長性をもたせて、信頼性を高めている。システム構成装置であるブート対象ディスク装置に格納されたシステムソフトウェアの処理によって冗長性を実現している。そのため、システムの立ち上げ処理において、一度、システムソフトウェアの起動まで処理が進んだ後は、以降の処理では、2重化しているシステムディスク装置へのアクセスにおいてエラーが発生しても、エラーの発生していない方のデータを用いて継続運用が可能となっている。

【0014】

また、2重化されたうちの主系のシステムディスク装置でエラーが多発した場合には、主系のディスク装置を切り離し、その後は、従系のシステムディスク装置からシステムをブートすることも可能である。しかし、システムの立ち上げ処理においては、ブート可能なディスク装置を2重化しても、その切り換えは、主系のディスク装置の異常を検出した場合をトリガにしている。そのため、主系のディスク装置に格納されたシステムソフトウェアの起動が進む前に、主系のシステムディスク装置で異常が発生した場合には、2重化の制御を行うシステムソフトウェアを動作させることができない。従って、システムディスク装置が二重化されていても、主系のシステムディスク装置に異常が発生すると、従系のシステムディスク装置に切り換えることができないため、システムを立ち上げることができないという問題が発生していた。

【0015】

そこで、本発明は、システムソフトウェアが起動後の装置異常の検出によるブート対象装置の切り換え処理という従来の機能に加えて、システムソフトウェアの起動前に装置異常が発生した場合にも、ブート対象装置を切り換えることができる機能を持たせた情報処理システムを提供し、更に、装置の異常検出とは、無関係に、ブート処理ごとに、強制的にブート対象装置を切り換える機能と、システムソフトウェアが起動時に強制的に切り換えられたブート対象装置を元に戻す処理を行う機能を有する情報処理システムを提供することを目的とする。

【0016】**【課題を解決するための手段】**

以上の課題を解決するため、本発明では、多重化された複数の装置から選択して読み出されたオペレーションシステムが起動されてシステムが立ち上げられる情報処理システムにおいて、前記複数の装置からブート対象を設定する環境情報を格納する格納手段と、前記環境情報の設定に基づいてブート対象装置を決定し、該ブート対象装置に格納された前記オペレーションシステムを起動させるブート制御手段と、前記複数の装置の多重化制御を実行する制御手段とを備え、前記ブート対象装置に異常が検出されたとき、前記環境情報の設定を変更して、他の

前記装置に切り換え制御することとした。

【0017】

そして、前記環境情報が、前記複数の装置についてブート対象候補とする装置設定情報を含む第1変数情報と、前記装置設定情報に基づいてブート対象装置を指定するインデックス情報を含む第2変数情報と、前記多重化が有効か否かの2値が設定される第3変数情報とを含むこととした。

【0018】

さらに、前記ブート制御手段は、システムの電源投入時において、前記インデックス情報を初期値にクリアし、前記装置設定情報に最初に設定された前記装置をブート対象として選択することとし、第3変数情報に有効が設定されている場合に、前記インデックス情報により指定されたブート対象装置を選択し、否が設定されている場合には、前記装置設定情報において最初に設定された装置をブート対象として選定することとした。

【0019】

さらに、前記ブート制御手段は、第3変数情報に有効が設定されている場合に、前記インデックス情報におけるインデックスを更新することとし、前記環境情報に設定されたブート対象装置を選定し、該ブート対象装置に格納されている前記オペレーションシステムを読み込み、起動させることとした。

【0020】

また、前記制御手段は、第3変数情報に有効が設定され、前記インデックス情報が初期値ではない場合には、第1変数情報における設定を参照して、従系のブート対象装置で立ち上がっているかどうかを認識し、従系のブート対象装置で立ち上がっていることを認識した場合には、主系の前記装置に異常が発生していることを示す警告メッセージを通知することとした。

【0021】

さらに、前記制御手段は、主系のブート対象装置で立ち上がっているか従系のブート対象装置で立ち上がっているかに関係なく、第2変数情報を初期値にクリアすることとした。

【0022】

前記制御手段は、第3変数情報が前記格納手段内に存在するかしないかに関係なく、前記ブート対象装置に異常の発生を検出したとき、当該ブート対象装置を切り離し、第2変数情報の前記ブート対象候補とする装置の設定情報を書き換えることとした。

【0023】

【発明の実施の形態】

以下において、本発明による情報処理システムの実施形態について、図を参照しながら説明する。しかし、本実施形態の情報処理システムを説明する前に、本実施形態における特徴と効果を明確なものとするために、本実施形態の情報処理システムの基礎となる従来技術による情報処理システムについて説明する。

【0024】

図7に、ブート・ブロック・プログラムとシステムソフトウェアとが独立にブート対象ディスク装置を切り換える従来技術による情報処理システムを示した。なお、同図は、情報処理システムの立ち上げに関連する部分を中心にして、その概略を示している。同図中において、図6に示した情報処理システムと同じ部分には、同じ符号が付されている。

【0025】

図7の情報処理システムでは、制御装置1に2重化されたディスク装置D1とD2が接続されている。このディスク装置D1とD2の夫々には、ディスクの先頭から、順に、ブート・ブロック・プログラム、OSローダ、OS、システムソフトウェアが格納されている。各ディスク装置には、図示していないが、これらに続いて、データが格納されている。

【0026】

制御装置1には、不揮発性メモリNMとメインメモリMとが備えられているが、図では、システム立ち上げ時に関連する部分が代表的に示されている。不揮発性メモリNMには、自己診断ファームウェアNM1、ブートファームウェア環境変数NM2が格納されている。図示していないが、この他にも、ブートファームウェアのバイナリデータが格納されている。ブートファームウェア環境変数NM2には、ブートファームウェアの環境情報が格納されており、図では、ディスク

装置D1とD2とで2重化されているので、その環境変数として、「boot-device=disk0, disk1」が格納されている。この環境変数において、「disk0」は、ディスク装置D1に、そして、「disk1」は、ディスク装置D2に対応付けられ、ブート対象が、ディスク装置D1とD2であり、立ち上げ時には、ディスク装置D1から起動処理が行われることを表している。

【0027】

また、メインメモリMには、システム立ち上げ前においては、何も格納されていないが、図においては、システム立ち上げ時にブート対象ディスク装置から読み込まれたソフトウェアの展開状況が示されている。ブートファームウェアM1は、不揮発性メモリNMに格納されていたブートファームウェアのデータを読み込んだものであり、ブート・ブロック・プログラム、OSロードM3、OSM4、システムソフトウェアM5は、ブート対象ディスク装置に格納されていたものが、システム立ち上げ時にメインメモリM内に読み込まれたものである。

【0028】

そこで、図7に示された情報処理システムにおいて、ブートファームウェアM1とシステムソフトウェアM5とが独立にブート対象ディスク装置を切り換える動作の概要を、次に示す。

【0029】

先ず、電源投入されて、システムの立ち上げ処理が開始されると、自己診断ファームウェアNM1が起動され、システムチェックが実行される。そして、不揮発性メモリNMに格納されているブートファームウェアのデータをメインメモリMに読み込み、ブートファームウェアM1とする。ブートファームウェア環境変数NM2に用意されているブート制御用の環境変数は、「boot-device=disk0, disk1」であるので、ブートファームウェアM1は、この環境変数に設定されたディスク装置を順番にアクセスし、最初に正常にアクセスできた装置をブート対象装置に決定する。ここで、ブート対象装置を決定すると、以降は、ブートファームウェアM1は、積極的にブート対象ディスク装置を変更することはない。

【0030】

次いで、ブートファームウェアM1は、上で選択したブート対象ディスク装置

にアクセスし、その先頭のブロックに書き込まれているブート・ブロック・プログラムをメインメモリに読み込み、ブート・ブロック・プログラムM2を展開する。ブートファームウェアM1は、読み込んだブート・ブロック・プログラムM2に制御を移す。

【0031】

ブート・ブロック・プログラムM2が起動すると、ブート・ブロック・プログラムM2は、自身が格納されていたディスク装置、図示の例では、ディスク装置D1内に格納されているOSローダをメインメモリに読み込み、OSローダM3に制御を移す。

【0032】

そこで、OSローダM3が起動すると、OSローダM3は、自身が格納されていたディスク装置D1に格納されているOSをメインメモリに読み込み、OSM4に制御を渡す。

【0033】

さらに、OSM4が起動すると、OSM4は、自身が格納されていたディスク装置D1に格納されているシステムソフトウェアをメインメモリM内に読み込み、システムソフトウェアM5を展開し、システムソフトウェアM5を起動する。ここで、システムソフトウェアM5が正常に起動した場合には、そのまま運用動作に移る。

【0034】

以上のように、図7に示された情報処理システムでは、ブートファームウェアM1とシステムソフトウェアM5とが独立にブート対象ディスク装置を切り換える動作を行っている。

【0035】

システムソフトウェアM5が正常に運用動作に移行できた場合には、システムソフトウェアM5が切り換え制御を実行できるため、ブート対象ディスク装置D1の異常を検出すると、ディスク装置D1を切り離すことができる。その場合には、次のリブートで、ブート対象ディスク装置D1を切り換えるため、システムソフトウェアM5が不揮発性メモリNMにあるブートファームウェア環境変数N

M2の環境情報を書き換える。

【0036】

しかしながら、この様にブートファームウェアM1とシステムソフトウェアM5が独立して処理を進めているため、ブートファームウェアM1が、ブートファームウェアの環境情報に従って、ブート対象ディスク装置を選択した以後であって、システムソフトウェアM5に制御が移行する前の処理中に、何らかの異常が発生して、処理を継続できなかった場合には、システムソフトウェアM5がまだ起動していないので、ブートファームウェア環境変数NM2の環境情報を書き換えることができない。そのため、次のリブートでも、ブート対象ディスク装置を異常のない他のディスク装置に変更できないことになり、その結果、システムを立ち上げることができない。

【0037】

この様に、従来技術による情報処理システムでは、システムソフトウェアM5の起動による制御が行われる以前の処理中に何らかの異常が発生したとき、ブート対象ディスク装置を冗長化しているにも関わらず、システムが立ち上がらないという問題があった。

【0038】

更に、従来技術の情報処理システムでは、ディスク装置の異常を検出したモジュールが、個々にブートファームウェア環境変数の環境情報「boot-device」を書き換える処理によって、ブート対象のディスク装置を切り換えようとする、ブート・ブロック・プログラム、OSローダ、OSの初期化処理のすべてにわたって、ブート対象のディスク装置の冗長構成を意識した変更が必要になるが、一般的に、ブート・ブロック・プログラム、OSローダ、OSの初期化部は、特定の冗長構成システムを意識した変更を適用することが困難であるという問題があった。

【0039】

そこで、本実施形態の情報処理システムでは、従来の情報処理システムに備えられた切り換え機構による以下の2つの要件をそのまま継承できるようにする。つまり、一つの要件は、ブートファームウェアは、環境情報「boot-device」に

設定されたディスク装置を順番にアクセスし、最初に正常にアクセスできたディスク装置をブート対象ディスク装置に決定することであり、二つ目の要件は、システムソフトウェアが正常に運用動作に移行できた場合には、該システムソフトウェアが切り換え制御でき、ディスク装置の異常を検出すると、ブート対象ディスク装置を切り離すことができることである。

【0040】

以上の様な要求を満たす情報処理システムとするため、本実施形態では、不揮発性メモリ内にあるブートファームウェア環境変数に格納される環境情報として、ブート対象となるデバイス情報を設定できる第1変数「boot-device」、ブート対象の順番情報を示す第2変数「boot-device-index」、そして、ディスク装置を順番にアクセスして、最初に正常にアクセスできたディスク装置をブート対象とする決定機能が有効であるかどうかを示す第3変数「boot-device-index-available」を用意した。

【0041】

第1変数「boot-device」は、冗長化されたディスク装置が、例えば、3つある場合には、「boot-device=disk0, disk1, disk5」のように設定される。また、第2変数「boot-device-index」については、例えば、第1変数が上記のようであれば、「disk0」のインデックス番号は、初期値0であり、「disk1」のインデックス番号は、1であり、「disk5」のインデックス番号は、2となり、次のブートで、ブート対象とするディスク装置を指定する。第3変数「boot-device-index-available」は、「true」と「false」の値を有しており、この値が選択されて、上述の2重化による切換機能が有効のときには、「true」が設定され、否のときには、「false」が設定される。

【0042】

以上のような3つの環境情報からなるブートファームウェア環境変数が不揮発性メモリに用意された本実施形態の情報処理システムの構成を図1に示した。図1では、図7に示した従来技術による情報処理システムの構成と同様の部分には、同じ符号が付されている。なお、図1においても、図7と同様に、情報処理システムの構成の中で、システム立ち上げに関連する構成を中心に示してある。

【0043】

図1の情報処理システムが、図7の情報処理システムと大きく異なることは、不揮発性メモリNMに格納されているブートファームウェア環境変数に用意されている内容である。図7の従来技術による情報処理システムに用意されているブートファームウェア環境変数NM2は、環境情報として、「boot-device」だけであったが、図1の本実施形態の情報処理システムでは、ブートファームウェア環境変数NM21に、環境情報として、第1変数「boot-device」、第2変数「boot-device-index」、そして、第3変数「boot-device-index-available」が用意されていることである。

【0044】

そこで、図1に示された本実施形態の情報処理システムは、ブートファームウェアとシステムソフトウェアによりブート対象ディスク装置を切り換える場合の概略構成を示し、2重化されたデバイスであるディスク装置を切り換える場合を例にしている。図2の処理手順のフロー図を参照しながら、その切り換え処理を以下に説明する。

【0045】

先ず、システムが、電源投入され、或いは、リブートリセットされたとき（ステップS101）、システムは、不揮発性メモリNM、例えば、フラッシュメモリ内に格納された、自身のハードウェア診断を実行するための自己診断ファームウェア（POST）NM1を起動させ、電源投入直後やリブートリセット直後にシステムが正常に立ち上がることを保証するために、ハードウェアの自己診断を実施する（ステップS102）。

【0046】

自己診断ファームウェアNM1は、メインメモリMを含めたシステムの診断を実行するため、通常は、フラッシュメモリ上のコードでそのまま動作する。ただし、メインメモリMの診断後は、メインメモリMにコードを展開して、メインメモリ上で動作する場合もある。また、CPUのキャッシュメモリの診断後に、キャッシュメモリに実行コードを展開して動作する場合もある。自己診断ファームウェアNM1は、ハードウェアを診断した結果として、ハードウェアの構成情報

を作成し、その情報を、後述のブートファームウェアM1に引き渡す。

【0047】

自己診断ファームウェアNM1による診断が終了するとブートファームウェアM1に制御を移す（ステップS103）。ブートファームウェアM1のコードは、通常、不揮発性メモリNM、例えば、フラッシュメモリなどに格納され、メインメモリMに実行コードが展開され、メインメモリM上で動作する。また、そのコードが、フラッシュメモリに格納されてなく、自己診断ファームウェアNM1により、外部からメインメモリMにダウンロードされる場合もある。

【0048】

ブートファームウェアM1は、自己診断ファームウェアNM1により作成されたハードウェア構成情報に基づいて、ブート対象プログラムであるブート・ブロック・プログラム、OSローダ、OSがブート処理を行えるように、ディスクドライバ、コンソールドライバなどブートドライバを初期化する（ステップS103）。また、ブートファームウェアM1は、ブート対象プログラムが動作されるように、メモリ環境などの環境を構築する。

【0049】

次いで、ブートファームウェアM1は、初期化処理の一環で、リセット要因をチェックし、リセット要因が、電源投入であった場合には、不揮発性メモリNMのブートファームウェア環境変数NM21に保存されている第2変数「boot-device-index」を0クリアする。リセット要因が、電源投入でないリブートである場合には、第2変数の値は、変更されない。

【0050】

ブート処理に進めるまで処理が進むと、ブートファームウェアM1は、ブートファームウェア環境変数NM21に保存されている3つの変数、第1変数「boot-device」、第2変数「boot-device-index」、第3変数「boot-device-index-available」の値に応じて、ブート対象ディスク装置を選択する（ステップS104）。

【0051】

ブートファームウェアM1は、選択したブート対象ディスク装置にアクセスし

て、ディスクの構成情報を読み込み、正常にディスクが動作することを確認する。選択したディスクが正常に動作できない場合には、ブート対象ディスク装置を次の候補のものに変更する。その際、第2変数「boot-device-index」の値を更新する。また、選択したブート対象ディスク装置が、正常に動作できた場合も、システムソフトウェアM5が正常に起動できずに、再度のブート処理されるときに備えるものとして、強制的に次の候補に切り換えるために、第2変数「boot-device-index」の値を更新する。

【0052】

なお、ブートファームウェア環境変数NM21に保存されている3つの変数「boot-device」、「boot-device-index」、「boot-device-index-available」の値から対象ディスクを選択する際に、選択できる対象ディスク装置がなかった場合、或いは、対象ディスク装置を次の候補のものに変更しようとしたときに、選択できる対象ディスク装置がなかった場合には、エラーメッセージを通知して、処理を停止する。

【0053】

ブートファームウェアM1は、ブート対象として選択されたのがディスク装置D1である場合には、該ディスク装置D1における先頭のブロックに書き込まれているブート・ブロック・プログラムをメインメモリMに読み込み、読み込んだブート・ブロック・プログラムM2に制御を移す（ステップS105）。

【0054】

ブート・ブロック・プログラムM2が起動すると、ブート・ブロック・プログラムM2は、自身が格納されていたディスク装置D1内に格納されているOSローダをメインメモリMに読み込み、OSローダM3に制御を移す（ステップS106）。

【0055】

OSローダM3が起動すると、OSローダM3は、自身が格納されていたディスク装置D1に格納されているOSをメインメモリMに読み込み、OSM4に制御が渡される（ステップS107）。

【0056】

そこで、OSM4の初期化処理が実施され、OSM4が起動すると（ステップS108）、OSM4は、自身が格納されていたディスク装置D1に格納されているシステムソフトウェアをメインメモリMに読み込み、システムソフトウェアM5を起動する（ステップS109）。

【0057】

そして、システムソフトウェアM5が正常に起動した場合には、システムソフトウェアM5が、ブートファームウェア環境変数NM21に記録された第2変数「boot-disk-unit-index」を0クリアする。この処理により、次のリブート処理でも、再度同じディスク装置D1から上述のブート処理が実行される。主系又は従系のどちらのディスク装置で立ち上がったかをチェックし、従系のディスク装置で立ち上がったと認識した場合には、エラーメッセージを通知するようにする（ステップS110）。以降のディスク装置の切り換え処理は、システムソフトウェアM5によって実施される（ステップS111）。

【0058】

ここで、システムソフトウェアM5が正常に起動できなかった場合には、ブートファームウェア環境変数NM21に記録された第2変数「boot-device-index」は、上述のステップS104における処理により、更新されたままの状態であるため、次のリブート時には、強制的に、次の候補のディスク装置がブート対象装置となる。また、ステップS104における処理で選択されたディスク装置に異常が検出され、その分余分に第2変数「boot-device-index」を更新済みであるため、次のリブート時には、強制的に、次の候補のディスク装置がブート対象となる。

【0059】

なお、ステップS4において、ブート対象ディスク装置のオープン処理で、該ディスク装置の異常を検出した場合には、上述したように、従来技術の情報処理システムと同様に、異常が検出された当該ディスク装置を切り離し処理が行われる。

【0060】

そして、システムソフトウェアM5の初期化処理に到達する以前のステップS

105乃至S109の処理期間中に、ディスク装置に異常などが発生し、パニック状態にあるとき、従来技術の情報処理システムでは、システムソフトウェアM5が起動することがないため、ディスク装置の切り換え処理が実行されなかった。しかし、本実施形態の情報処理システムでは、システムソフトウェアM5が正常に起動できていないことを認識でき、強制的にディスク装置を切り換えることができる。2重化されたうちの主系のシステムディスク装置で異常が発生した場合、2重化の制御を行うシステムソフトウェアが動作していないため、従系のシステムディスク装置に切り替えることができずにシステムを立ち上げることができないという問題を回避できる。

【0061】

このように、図1に示された情報処理システムにおいて、2重化されたディスク装置に対して、ブートファームウェアとシステムソフトウェアによりブート対象ディスク装置を切り換える処理手順が説明された。次いで、図3及び図4を参照して、図1の本実施形態の情報処理システムにおけるブートファームウェアM5の詳細構成と、ブートファームウェアM5によるブート対象ディスク装置の切り換え処理の手順を、以下に説明する。

【0062】

図3は、図1に示された本実施形態の情報処理システムの制御装置1内に備えられた不揮発性メモリNMの格納状況と、メインメモリM内に読み込まれたブートファームウェアM1の構成の具体例を示している。図3において、図1と同様の部分には、同じ符号が付されている。

【0063】

制御装置1内に備えられた不揮発性メモリNMには、自己診断ファームウェア(POST)NM1、ブートファーストウェアの圧縮データNM3、ハードウェア構成情報NM4が格納されているが、これらは、従来技術の情報処理システムの不揮発性メモリNMにおいても、用意されているものである。従来のものにおける不揮発性メモリNMに格納されていたブートファームウェア環境変数NM2の代わりに、図3のブートファームウェア環境変数NM21には、環境情報として、第1変数「boot-device」、第2変数「boot-device-index」、第3変数「bo

ot-device-index-available」が用意されている。

【0064】

一方、ブートファームウェアM1は、従来技術によるブートファームウェアM1の構成と同様であり、初期化処理部M11、トラップ処理部M12、メモリ処理部M13、モニタ処理部M14、各種ライブラリ部M15、クライアントサービス処理部M16、ブート処理部M17、そしてドライバ処理部M18のモジュールから構成される。これらのモジュール構成を有するブートファームウェアM1は、システムの自己診断完了後において、ブートファームウェアのデータNM3からメインメモリM内に読み込まれ、圧縮されて保存されていたものを伸張されて展開される。ブートファームウェアM1は、メインメモリM上で動作する。これらの各部は、次のようである。

【0065】

初期化処理部M11は、ブートファームウェアM1の各モジュールの初期化を行う。不揮発性メモリNMのブートファームウェア環境変数NM21に格納された環境情報において、初期化の必要なものは、ここで初期化される。

【0066】

トラップ処理部M12は、ブートファームウェアM1の動作環境で発生するトラップの処理を実行する。トラップ処理のなかには、ハードウェアの障害が原因で発生するものもあれば、メモリ管理ユニット（MMU）におけるミス処理のように、通常処理の一部として発生するものもある。また、OSのブート処理の過程で、メインメモリM上に展開されて動作するブート・ブロック・プログラムM2、OSローダM3は、トラップベクタがブートファームウェアM1の環境で動作するため、それらの動作時に発生するトラップの処理も、トラップ処理部M12が行う。また、メインメモリMに展開されたOSM4についても、初期のトラップベクタ変更前までのトラップ処理は、トラップ処理部M12によって行われる。

【0067】

メモリ管理部M13は、不揮発性メモリNMのハードウェア構成情報NM4で通知された物理メモリの構成から論理アドレスの割り当てを行う。また、MMU

のミス処理に伴う動的なメモリの割り当てを行う。また、物理メモリ、論理メモリの動的な獲得／解放処理を行う。

【0068】

モニタ処理部M14は、ブートファームウェアM1のマン・マシン・インタフェースであるコマンド入力画面をサポートし、各種ライブラリ部M15は、いろいろなサポートモジュールである。

【0069】

クライアントサービス処理部M16は、メインメモリM上に展開されたブート・ブロック・プログラムM2や、OSローダM3や、OSM4などのクライアント・プログラムに対する動的なメモリ空間の割り当てや、ディスク装置へのアクセスなどのサービスを提供するインタフェースの役割を有している。各クライアント・プログラムは、クライアントサービス処理部M16を通して、ブートファームウェアM1の有する機能を利用することができる。この機能を利用して、ブート対象ディスク装置に格納されているブート・ブロック・プログラム、OSローダ、OSを読み出し、ブート処理が可能になる。

【0070】

ブート処理部M17は、ブート処理をサポートするモジュールである。不揮発性メモリNMのブートファームウェア環境変数NM21に格納された環境情報の値に応じて、ブート対象ディスク装置を決定し、ブート対象ディスク装置の先頭ブロックに格納されているブート・ブロック・プログラムをメインメモリMに展開し、制御を渡す機能を持つ。本実施形態の特徴であるステップS104の処理を行うモジュールである。

【0071】

ドライバ処理部M18は、ブートファームウェアM1がサポートするI/Oを制御するものである。これには、ディスクドライバやSCSIドライバ、コンソールドライバなどが含まれる。

【0072】

以上のような各種モジュールからなるブートファームウェアM1がメインメモリM上での動作について、図4の処理フローを参照して説明する。

【0073】

先ず、システムは、電源投入直後又はリブートリセット直後は（ステップS201）、不揮発性メモリであるフラッシュメモリに書き込まれた自己診断ファームウェア（POST）を起動する（ステップS202）。

【0074】

自己診断ファームウェアNM1は、システムが正常に立ち上がることを保証するためのハードウェア診断を実行する。そして、診断の結果として、ハードウェアの構成情報を作成し（ステップS203）、不揮発性メモリNMのハードウェア構成情報NM4に記録する（ステップS204）。この構成情報は、その後のブートファームウェアM1でのハードウェア構成認識に使用される。

【0075】

ブートファームウェアM1は、自己診断ファームウェアNM1によるハードウェア診断の後で、メインメモリMを診断した後で起動するプログラムであるため、メインメモリM上で動作することができる。ブートファームウェアM1のバイナリデータは、通常は、システム内のフラッシュメモリに圧縮された形式で格納されており、POSTの診断後に、POSTによりメインメモリM上に伸張して展開される（ステップS205）。或いは、ブートファームウェアM1のデータについては、システム内のフラッシュメモリに格納せず、外部からPOSTにより圧縮されたデータをメインメモリM上に伸張してダウンロードするようにしてもよい。

【0076】

そして、図2に示されたステップS103の処理と同様に、ブートファームウェアM1の各モジュールM11乃至M18の初期化を実行し、さらに、初期化処理部M11が、ハードウェア構成情報NM4に従って、ブートファームウェア環境変数NM21に従う初期化を実行する（ステップS206）。

【0077】

ブート処理部M17が、ブートファームウェア環境変数NM21の環境情報に従って、ブート対象ディスク装置を決定する（ステップS207）。そこで、ブート処理部M17は、決定したブート対象のディスク装置にアクセスし、当該デ

ディスク装置の先頭ブロックに格納されているブート・ブロック・プログラムを読み出し、メインメモリM上に展開する（ステップS208）。ブート・ブロック・プログラムM2は、OSローダをメインメモリに展開して制御を渡す機能を持つ。

【0078】

次いで、メインメモリM上に読み込まれたブート・ブロック・プログラムM2は、クライアントサービス処理部M16の機能を利用して、ブート対象ディスク装置の先頭にあるブート・ブロック・プログラムの次に格納されているOSローダを読み出し、メインメモリM上に展開する（ステップS209）。OSローダM3は、OSのファイルシステムを解釈できる機能を持ち、ブート対象ディスク装置に格納されたOSプログラムをメインメモリに展開して制御を渡す機能を持つ。

【0079】

さらに、メインメモリM上に読み込まれたOSローダM3は、クライアントサービス処理部M16の機能を利用して、ブート対象ディスク装置に格納されたOSローダの次のOS読み出し、メインメモリM上に展開する（ステップS210）。この処理は、図2に示した処理フローのステップS108と同様であり、ステップS210以降の処理は、ステップS109以降の処理に同じである。

【0080】

以上では、本実施形態の情報処理システムにおけるブートファームウェアM5の詳細構成と、ブートファームウェアM5によるブート対象ディスク装置の切り換え処理の手順について説明したが、次に、図5のフローチャートを参照して、本実施形態の情報処理システムにおける制御装置1内のメインメモリMに、ブート対象のディスク装置に格納されているシステムソフトウェアが読み出されて展開された以降におけるディスク装置の切り換え処理について、説明する。

【0081】

まず、メインメモリM上に読み込まれたシステムソフトウェアM5が、ステップS110で説明されたように、初期化処理されて起動される（ステップS301）。

【0082】

次いで、システムソフトウェアM5は、不揮発性メモリNMのブートファームウェア環境変数NM21を調べ、第3変数「boot-device-index-available」が存在するかチェックする（ステップS302）。

【0083】

ここで、本実施形態を実施する以前の、従来技術の情報処理システムの構成によるブートファームウェアM5でブート処理を実行した場合には、図7で示したように、ブートファームウェア環境変数NM2には、第3変数「boot-device-index-available」は、存在していないので（ステップS302のN）、図7で示した従来技術のシステム構成の範囲内で動作するため、直ちに、OSの通常運用処理に移行する（ステップS305）。

【0084】

本実施形態を実施している場合には、ステップS302の処理において、第3変数「boot-device-index-available」が存在し（Y）、現在ブート対象ディスク装置の冗長運用機能が設定されているかどうか確認する（ステップS303）。システムソフトウェアM5は、ブート対象ディスク装置を冗長運用することにより信頼性を高めるソフトウェアであるが、設定によりこの機能を無効にする場合が有り得るので、現在の設定を、ここで確認する。

【0085】

ステップS303における冗長運用機能に係る設定の確認により、システムソフトウェアM5がブート対象ディスク装置を冗長運用するように設定されていた場合には（Y）、ブートファームウェア環境変数NM21の第3変数「boot-device-index-available」に値「true」を設定する（ステップS306）。この第3変数「boot-device-index-available」は、最初のデフォルトの状態では、値「false」に設定されている。この場合には、ブートファームウェアM1での強制ディスク切り換え機能は働かない。

【0086】

しかし、一度、値が「true」に設定されると、この第3変数は不揮発性メモリNMのブートファームウェア環境変数NM21に記録されているため、誰かが意

図的に変更しなければ、その値は変わらない。もし、システムソフトウェアM5がインストールされていないときに、第3変数に「true」が設定されてしまうと、ブートファームウェアM1によるブート時に、強制的にブート対象ディスク装置を切り換えても、もとに戻すものがなくなってしまう。そのため、リブート処理に失敗することとなる。それを避けるため、第3変数の値を「true」にする設定は、システムソフトウェアM5で行うようになっている。

【0087】

システムソフトウェアM5がインストールされ、運用されているときには、一度、第3変数の値が「true」に設定されると、第3変数の値は、不揮発性メモリNMのブートファームウェア環境変数NM21に保持されるため、冗長運用の機能が有効となる。なお、この処理フローでは、システムソフトウェアM5の起動処理で、ブート対象ディスク装置の冗長運用状態をチェックして、必ず、第3変数を変更するとしているが、ここでは、変更を行わずに、システムソフトウェアM5のインストール時に設定するようにし、或いは、通常動作状態に入ってから、システムソフトウェアM5のメニューから、冗長運用状態をチェックし、設定してもいい。システムソフトウェアM5が、唯一、第3変数の値を「true」に設定するということが重要である。

【0088】

ステップS303における確認処理により、システムソフトウェアM5は、ブート対象ディスク装置が冗長運用するように設定されていなかった場合には(N)、ブートファームウェア環境変数NM21の第3変数「boot-device-index-available」に値「false」を設定する(ステップS304)。

【0089】

次に、ブートファームウェア環境変数NM21の第2変数「boot-device-index」が0かどうかを調べる(ステップS307)。図面2でのフローで示したように、第2変数の値が0というのは、ブートファームウェアM1の動作時に、第3変数「boot-device-index-available」の値が「false」であった場合だけである。ブートファームウェアM1の動作時に、第3変数「boot-disk-unit-index-available」の値が「true」であった場合には、ブート処理の実行で、必ず、第2

変数「boot-device-index」の値は更新(インクリメント)されている。従って、第2変数の値が0の場合には(ステップS307のY)、本実施形態による切り換え機能は有効となっていないので、直ちに、通常運用に移行する(ステップS305)。

【0090】

一方、第2変数「boot-device-index」の値が0でない場合は(ステップS307のN)、ブートファームウェアM1の動作時に、第3変数「boot-device-index-available」の値が「true」であったことを示している。この場合に、第1変数「boot-device」の設定をチェックし(ステップS308)、実際にブート処理が実施されたディスク装置が、主系のものであるか従系のものであるかをチェックする(ステップS309)。

【0091】

一般には、第2変数「boot-device-index」の値が1の場合には、主系で立ち上がっており、2以上の場合には、従系で立ち上がっていることを示している。システムの設定によっては、第1変数「boot-device」への設定で、同一のディスク装置を別名で複数設定する場合もある。その場合は、第2変数「boot-device-index」の値が2でも、主系の装置を示している場合もある。ここでは、そのような状態も考慮して確認する。

【0092】

ステップS309の処理における確認の結果、ブートしたディスク装置が主系でないと認識できた場合には(N)、主系のディスク装置の起動で、何らかの異常が発生しているため、それを通知する警告メッセージを表示する(ステップS310)。

【0093】

ブート起動した装置が主系であろうと従系であろうと、システムソフトウェアM5が正常に起動したことを示し、次回のリブート処理では、再度、正常にブート処理が実行できるように、第2変数「boot-device-index」の値を0クリアして(ステップS311)、通常運用に移行する(ステップS305)。なお、ここでは、0クリアとしているが、デクリメントして、最後にブートが成功した装

置を示すようにしてもよい。

【0094】

ステップS305において、通常運用に移行するが、通常運用の処理中に、ディスク装置の異常が検出された場合には、システムソフトウェアM5により、ブート対象ディスク装置が切り離され、次のブートでも、ブート対象ディスク装置を切り換えるために、システムソフトウェアM5が、環境情報の第2変数「boot-device-index」の値を、直接書き換える。この処理は、従来技術の場合と同様である。なお、このときに、ステップS311において、第2変数「boot-device-index」の値を0クリアせずに、デクリメントしていた場合には、第1変数「boot-device」の変更に合わせて、第2変数「boot-device-index」の値も書き換えるか、或いは、0クリアすることになる。

【0095】

以上のように、本実施形態による情報処理システムによれば、オペレーティングシステムを格納したシステムディスク装置であるブート対象とできるデバイス装置を2重化し、運用時のシステムディスク装置のアクセスで発生するハードエラーに対して冗長性を持たせて信頼性を向上できたコンピュータシステムとすることができ、ブートファームウェアと、ディスク装置の2重化処理のためのシステムソフトウェアとを連携させることにより、主系のブート対象デバイス装置に格納されたシステムソフトウェアが起動できなかった場合に、次のブートファームウェアによるブート処理において、ブート対象デバイス装置を強制的に従系のブートデバイス装置に切り換えることにより、従系のブートデバイス装置からのシステムソフトウェアの起動を可能とすることができた。

【0096】

以下に、本発明による情報処理システムに関する実施態様を示す。

(付記1)

多重化された複数の装置から選択して読み出されたオペレーションシステムが起動されてシステムが立ち上げられる情報処理システムにおいて、

前記複数の装置からブート対象を設定する環境情報を格納する格納手段と、

前記環境情報の設定に基づいてブート対象装置を決定し、該ブート対象装置に

格納された前記オペレーションシステムを起動させるブート制御手段と、

前記複数の装置の多重化制御を実施する制御手段と、を有し、

前記制御手段は、前記ブート対象装置に異常が検出されたとき、前記環境情報の設定を変更して、他の前記装置に切り換え制御することを特徴とする情報処理システム。

(付記 2)

前記環境情報は、

前記複数の装置についてブート対象候補とする装置設定情報を含む第 1 変数情報と、

前記装置設定情報に基づいてブート対象装置を指定するインデックス情報を含む第 2 変数情報と、

前記多重化が有効か否かの 2 値が設定される第 3 変数情報と、を含むことを特徴とする付記 1 に記載の情報処理システム。

(付記 3)

前記ブート制御手段は、システムの電源投入時において、前記インデックス情報を初期値にクリアし、前記装置設定情報に最初に設定された前記装置をブート対象として選択することを特徴とする付記 2 に記載の情報処理システム。

(付記 4)

前記ブート制御手段は、第 3 変数情報に有効が設定されている場合に、前記インデックス情報により指定されたブート対象装置を選択し、否が設定されている場合には、前記装置設定情報において最初に設定された装置をブート対象として選定することを特徴とする付記 2 に記載の情報処理システム。

(付記 5)

前記ブート制御手段は、第 3 変数情報に有効が設定されている場合に、前記インデックス情報におけるインデックスを更新することを特徴とする付記 4 に記載の情報処理システム。

(付記 6)

前記ブート制御手段は、前記環境情報に設定されたブート対象装置を選定し、該ブート対象装置に格納されている前記オペレーションシステムを読み込み、起

動させることを特徴とする付記 2 乃至 5 のいずれか一項に記載の情報処理システム。

(付記 7)

前記制御手段は、第 3 変数情報に有効が設定され、前記インデックス情報が初期値ではない場合には、第 1 変数情報における設定を参照して、従系のブート対象装置で立ち上がっていることを認識することを特徴とする付記 2 に記載の情報処理システム。

(付記 8)

前記制御手段は、従系のブート対象装置で立ち上がっていることを認識した場合には、主系の前記装置に異常が発生していることを示す警告メッセージを通知することを特徴とする付記 7 に記載の情報処理システム。

(付記 9)

前記制御手段は、第 3 変数情報が存在し、第 2 変数情報が 0 でない場合には、第 2 変数情報を初期値にクリアすることを特徴とする付記 7 又は 8 に記載の情報処理システム。

(付記 10)

前記制御手段は、第 3 変数情報が前記格納手段内に存在する場合であって、前記複数の装置に関して冗長運用の設定がないときには、第 3 変数情報を否に設定することを特徴とする付記 2 に記載の情報処理システム。

(付記 11)

前記制御手段は、第 3 変数情報が前記格納手段内に存在するかどうかに関係なく、前記ブート対象装置に異常の発生を検出したとき、当該ブート対象装置を切り離し、第 1 変数情報の前記ブート対象候補とする装置の設定情報を書き換えることを特徴とする付記 2 に記載の情報処理システム。

(付記 12)

前記格納手段は、不揮発性メモリであり、前記ブート制御手段及び前記制御手段によって、該格納手段に格納された第 2 変数情報及び第 3 変数情報における設定が書き換えられることを特徴とする付記 2 に記載の情報処理システム。

(付記 13)

前記ブート制御手段は、前記格納手段に格納されたブートファームウェアによってブート処理を実行することを特徴とする付記 1 又は 2 に記載の情報処理システム。

(付記 14)

前記制御手段は、前記ブート対象装置に格納されたシステムソフトウェアが読み込まれ、多重化制御を実行することを特徴とする付記 1 又は 2 に記載の情報処理システム。

【0097】

【発明の効果】

以上のように、本発明の情報処理システムでは、ブート対象装置であるシステムディスク装置を多重にして備え冗長化を向上させるシステムにおいて、多重化を制御するシステムソフトウェアが起動する前に、ブートファームウェアによって選択された主系のシステムディスク装置で異常が発生した場合でも、ブート対象装置を従系のシステムディスク装置に切り換えて運用することが可能となる。

【0098】

また、ブート対象装置の多重化制御をするシステムソフトウェアは、ブート対象装置内に格納された OS が起動した後に、OS 配下で動作するソフトウェアであり、OS 自体は、ブートローダにより起動される。本発明の情報処理システムでは、システムの不揮発性メモリに格納されるブートファームウェア環境情報の設定の仕方を工夫することによって、多重化を制御するシステムソフトウェアが起動する前に、主系のシステムディスク装置で異常が発生した場合でも、ブート対象装置を従系のシステムディスク装置に切り換えられるようにした。そのため、特に、OS やブートローダの変更が許されない場合でも、この OS とブートローダには、一切変更を加えないで、ブート対象装置の切り換えが実現できた。

【図面の簡単な説明】

【図 1】

2 重化されたブート対象デバイスを有する本実施形態による情報処理システムに係る主要部の概略ブロック構成を説明する図である。

【図 2】

本実施形態の情報処理システムのシステム起動時における処理手順を説明するフロー図である。

【図 3】

本実施形態による情報処理システムのシステム起動時におけるメインメモリへのブートファームウェアの展開を説明する図である。

【図 4】

本実施形態の情報処理システムのシステム起動時におけるブートファームウェアによる処理手順を説明するフロー図である。

【図 5】

本実施形態の情報処理システムのシステム起動時におけるシステムソフトウェアによる処理手順を説明するフローチャートである。

【図 6】

従来技術による情報処理システムの概略ブロック構成を説明する図である。

【図 7】

2重化されたブート対象デバイスを有する従来技術による情報処理システムに係る主要部の概略ブロック構成を説明する図である。

【符号の説明】

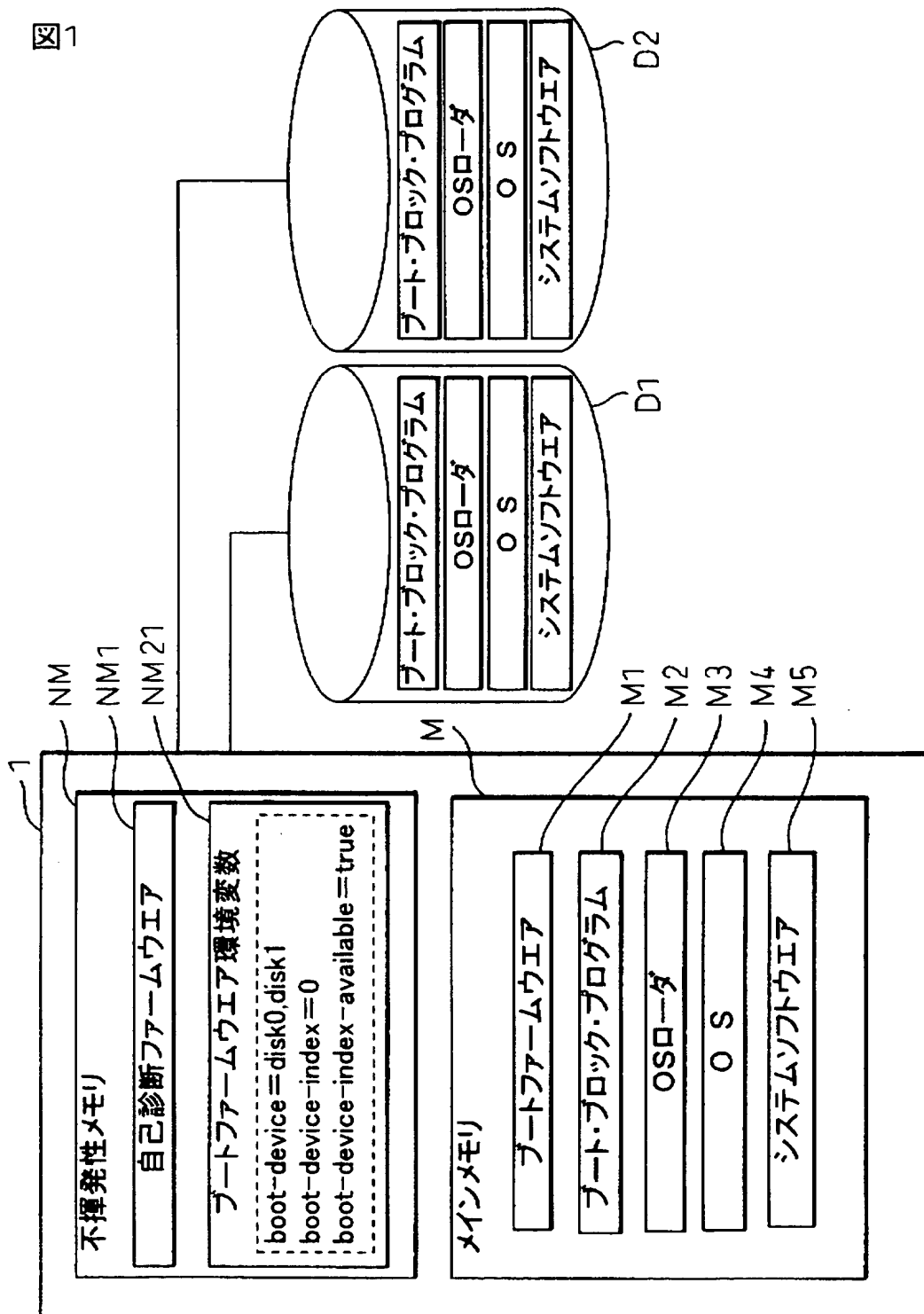
- 1…制御装置
- 2…制御部
- 3…不揮発性メモリ
- 4…メインメモリ
- 5…周辺装置
- D1、D2…ディスク装置

【書類名】

図面

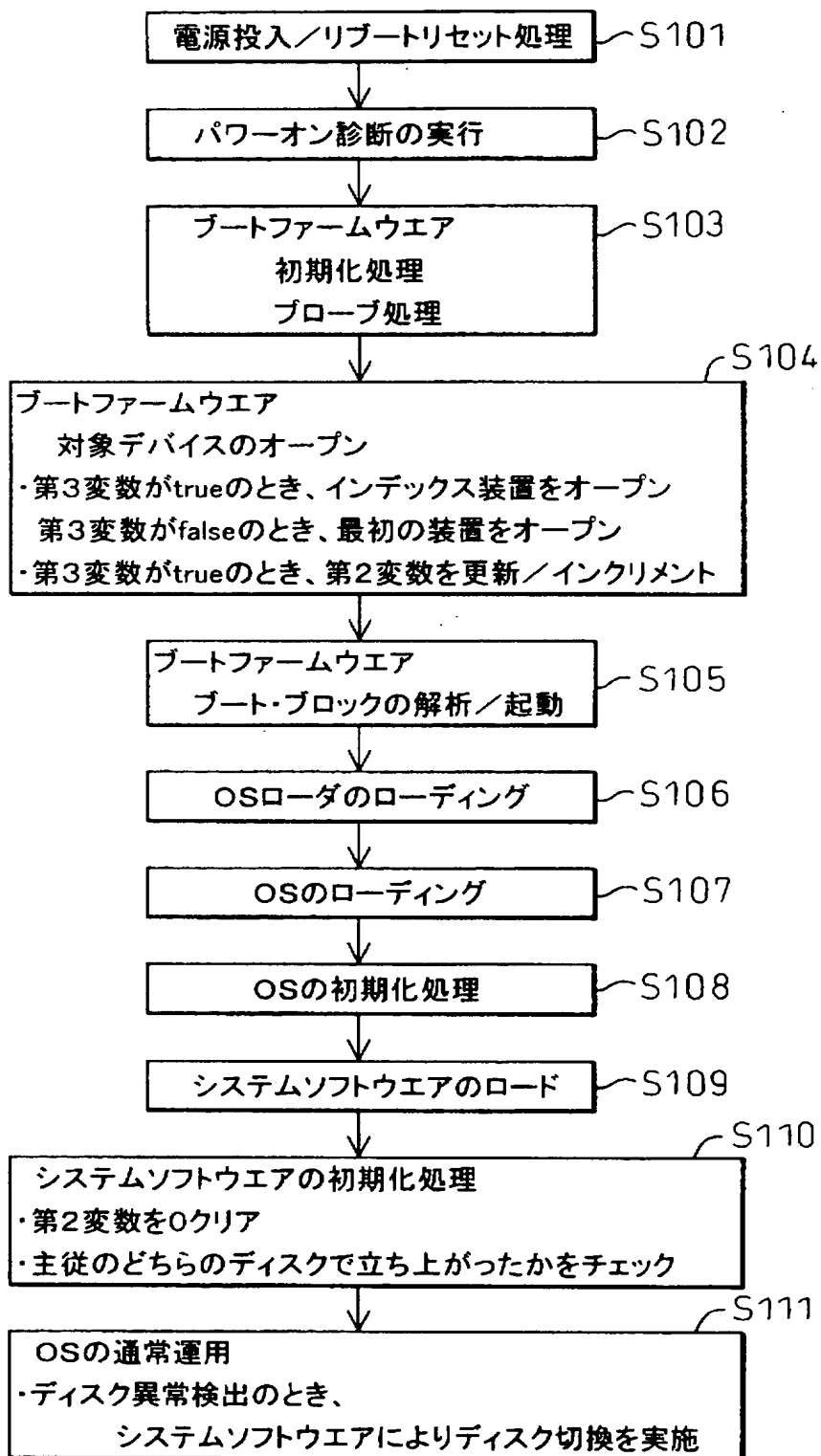
【図 1】

図1



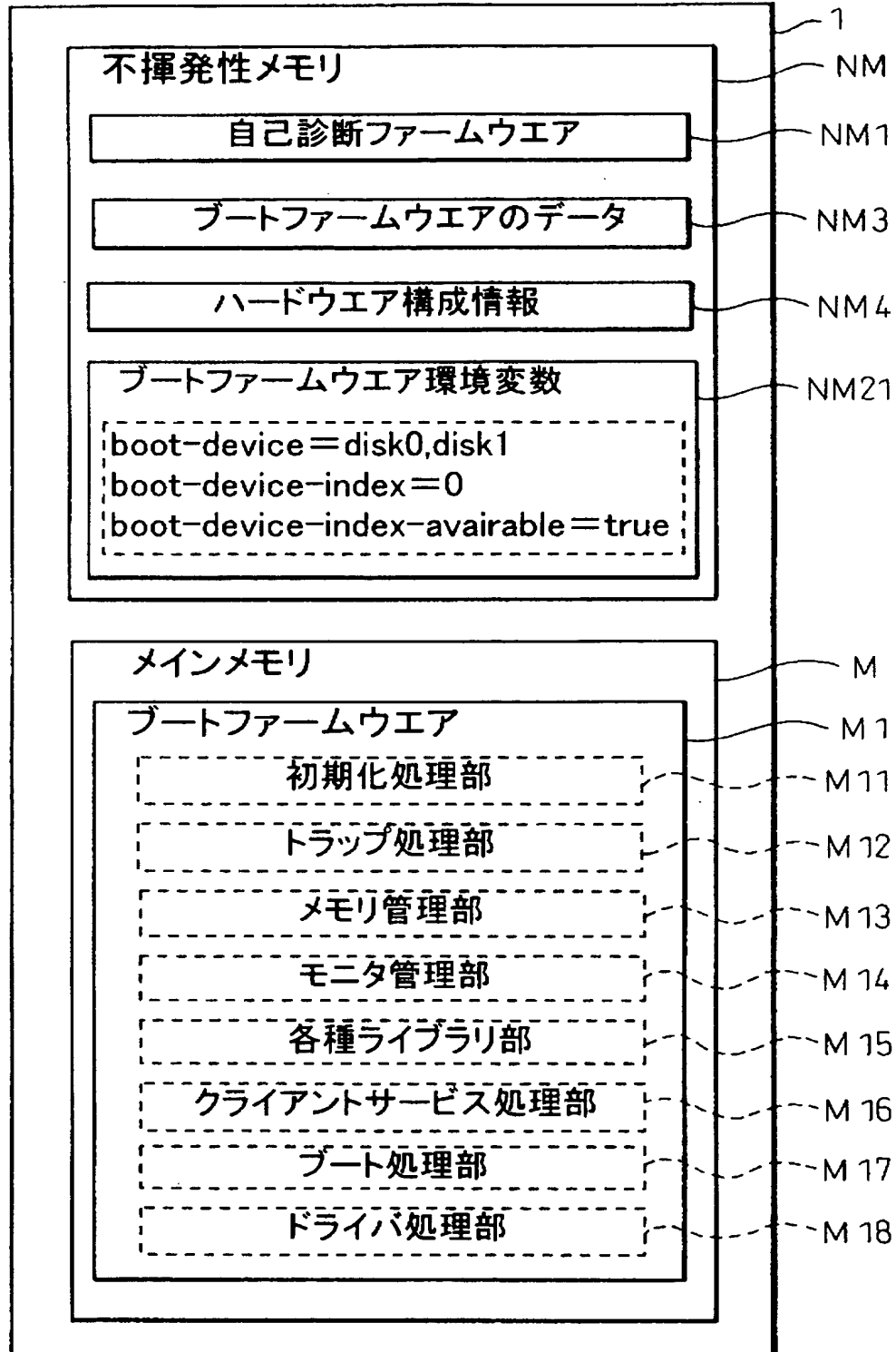
【図 2】

図 2



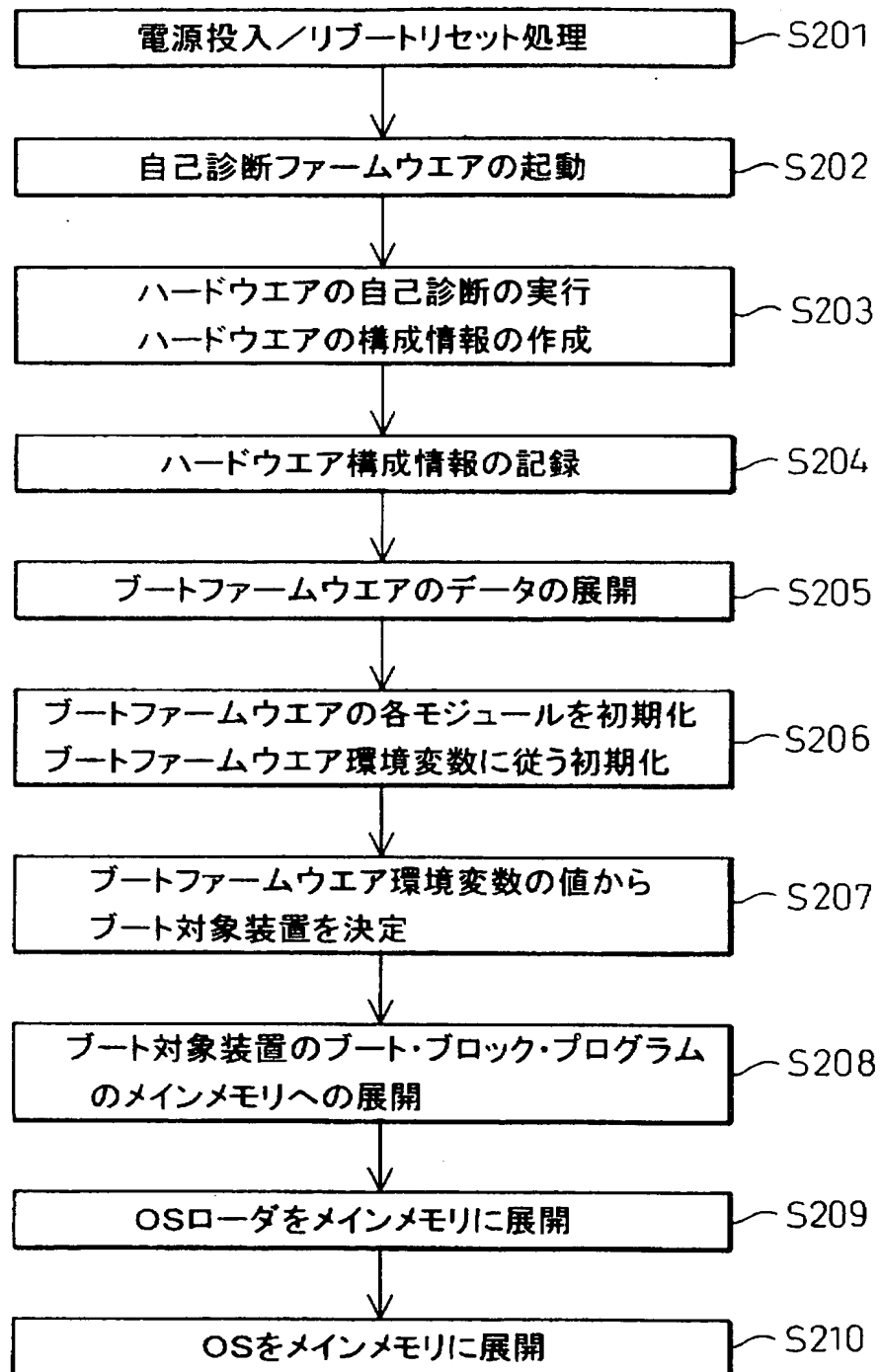
【図 3】

図 3



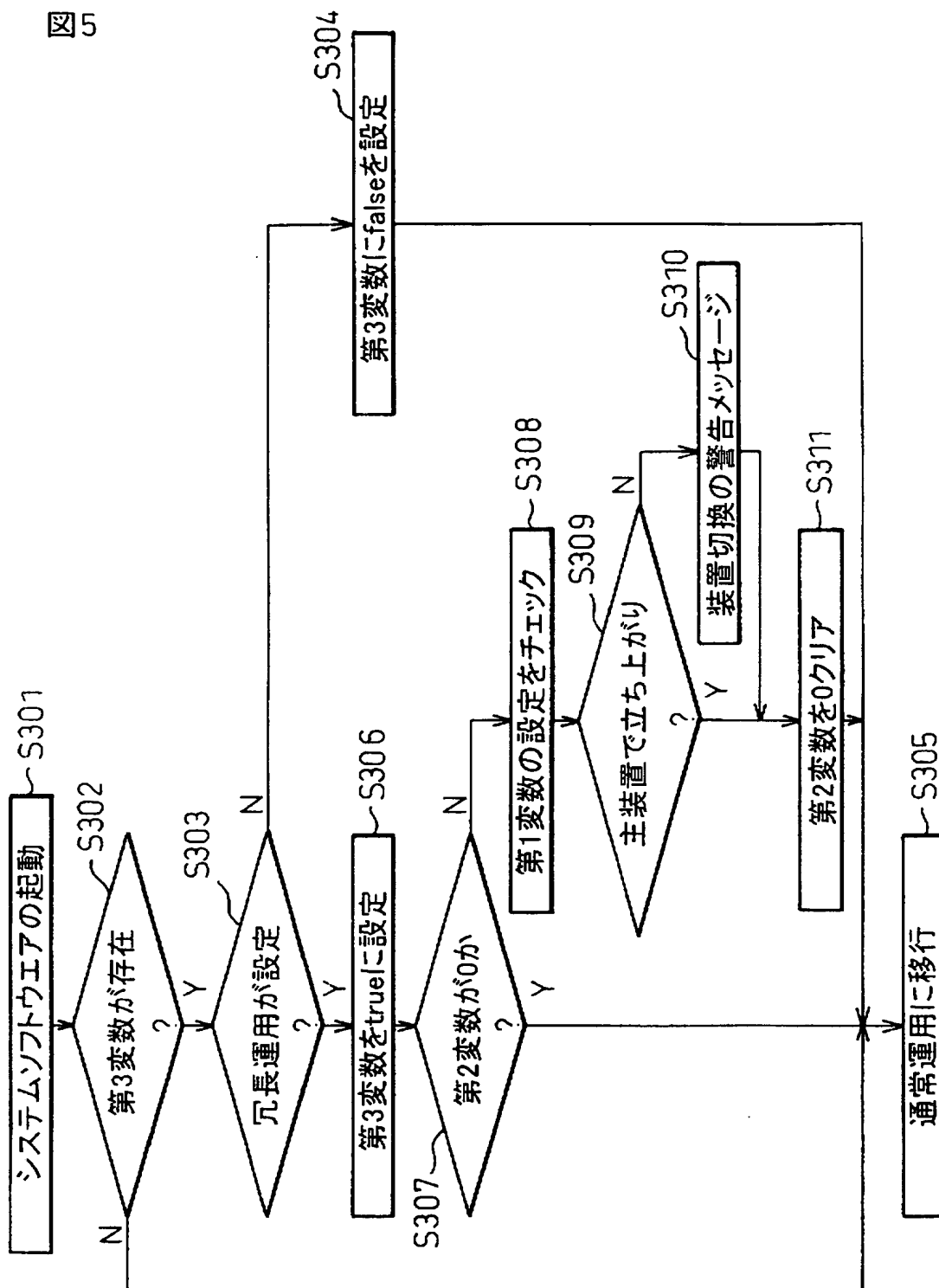
【図 4】

図 4



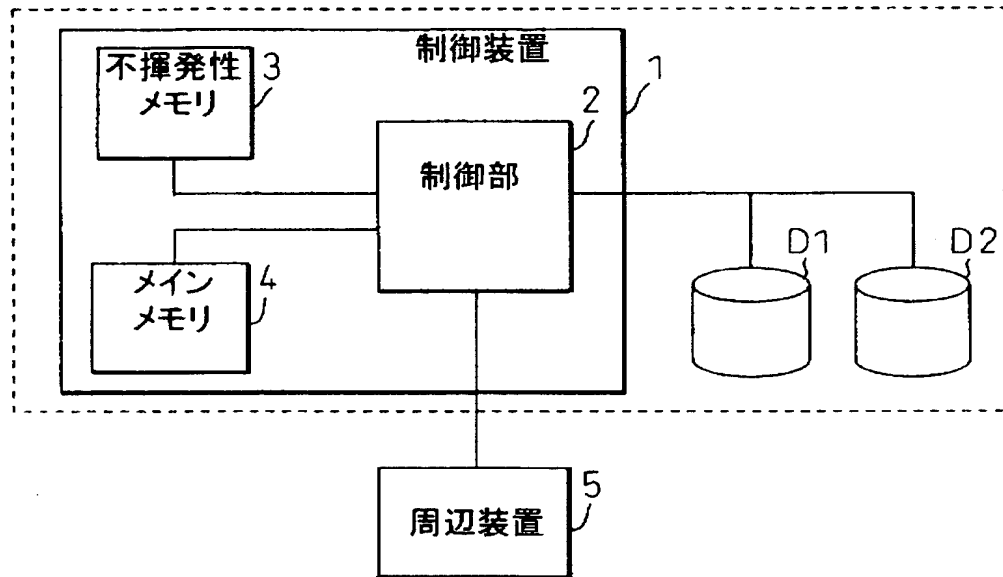
【図 5】

図 5



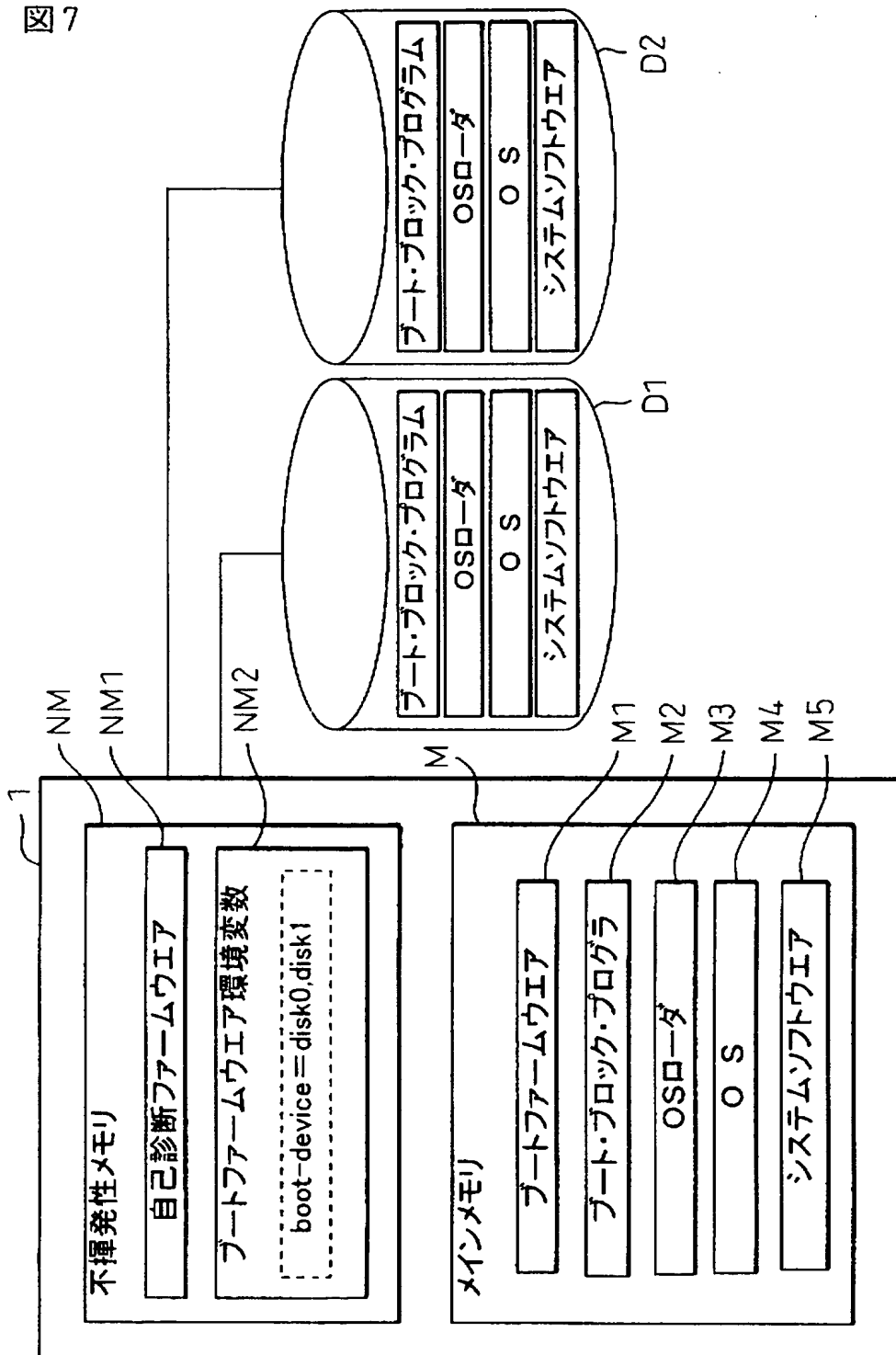
【図 6】

図 6



【図 7】

図 7



【書類名】 要約書

【要約】

【課題】 本発明は、OSを格納した装置を多重化して備え、主系のブート対象装置で起動できない場合に、強制的に従系のブート対象装置に切り換え、次のブート処理で従系のブート対象装置で起動する情報処理システムを提供する。

【解決手段】 制御装置1は、2重化装置D1,D2から選択し読み出したOSを起動してシステムを立ち上げる。不揮発性メモリNMには、ブート対象候補の装置設定情報を含む第1変数情報、該装置設定情報に基づきブート対象装置を指定するインデックス情報を含む第2変数情報、多重化が有効か否かの2値が設定される第3変数情報とを含むブートファームウェア環境変数NM21が格納される。ブートファームウェアM1は、環境変数の設定に基づきブート対象装置を決定し、ブート対象装置に格納されたOS M4を起動する。システムソフトウェアM5は、前記環境変数の設定に基づきブート対象装置の多重化制御を実施する。

【選択図】 図1

特願 2 0 0 3 - 1 1 9 8 0 8

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1. 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社